

PATENT APPLICATION

Log Analyzing Method, Execution Apparatus of the Same, and Recording Medium for Recording Process Program Thereof

Inventors: **Yumiko Seki**
Citizenship: Japan

Shinkichi Ebata
Citizenship: Japan

Assignee: **Hitachi, Ltd.**
6, Kanda Surugadai 4-chome
Chiyoda-ku, Tokyo, Japan
Incorporation: Japan

Entity: Large

- 1 -

LOG ANALYZING METHOD, EXECUTION APPARATUS OF THE SAME,
AND RECORDING MEDIUM FOR RECORDING PROCESS PROGRAM
THEREOF

5 BACKGROUND OF THE INVENTION

Field of the Invention

The present invention is related to a log
analyzing apparatus for analyzing log information which
is outputted by an application system in a computer.

10 More specifically, the present invention is directed to
such a technique effectively applied to such a log
analyzing apparatus capable of improving an application
program based upon an analysis result of log
information.

15

Description of the Related Art

Conventionally, in the case that errors
happen to occur while programs are executed by way of
computers, error messages which are outputted from
20 systems when the errors happen to occur are analyzed,
and/or log information outputted while programs are
executed are analyzed so as to investigate the reasons
why these errors occur. Thereafter, the programs are
improved, if necessary.

25

For instance, in a database process system,
an on-line system such as a system of a financial
institute, and an operation management system of a

railroad, while journal (system history) information is acquired as the above-explained log information during the system operations, this journal information is used to be investigated in order that these systems are
5 recovered when the failures occur.

It should be understood that the below-mentioned journal acquiring/analyzing apparatus is described in JP-A-2000-122901. That is, since the journal points are defined and used with respect to the
10 program, this journal acquiring/analyzing apparatus can easily set the journal information, and can readily compare/investigate the acquired journal information with the design data.

In the above-explained prior art, the log
15 information outputted when the program is executed is analyzed so as to check the occurring reason of the error. As a result, in such a case that the user does not know the internal structure of the program, the prior art system owns such a problem that the log
20 information cannot be analyzed, but also the program cannot be improved.

For instance, in such a case that while a user conducts such a process system as a database process system and produces a process program specific
25 to this user by employing a simplified language such as script, this user operates the user specific process program and an error happens to occur, since the user does not own specialized techniques similar to these

owned by system developing engineers, this user cannot perform the log analyzing operation. Even when a relatively simple mistake may constitute the occurring reason of the above-explained error, there is such a problem that lengthy time is necessarily required to investigate the reason why such an error happens to occur, and also lengthy time is necessarily needed so as to improve the process program.

10 SUMMARY OF THE INVENTION

The present invention has been made to solve the above-described problems, and therefore, has an object to provide such a technique capable of analyzing logs even in such a case that a user does not know an internal structure of a program. The present invention is featured by such a log analyzing apparatus for analyzing logs outputted when a program is executed by a computer, a reason why an error happens to occur while the program is executed by the computer by judging as to whether or not each of operations when such a program is executed, which are extracted from log information can satisfy an initial condition thereof.

In the log analyzing apparatus of the present invention, firstly, the log analyzing apparatus accepts as input information such a detailed log information which is outputted with respect to a certain range of a program, and then reads this accepted log information

so as to acquire an operation sequence of a predetermined application and also log level definition information. Then, the log analyzing apparatus forms a log table in which contents of the log information are
5 edited with respect to each of operation sequences of the application.

After a log designation condition indicative of log information to be extracted has been accepted from the user, the log analyzing apparatus extracts
10 information from the log table based upon this user designation condition, and judges as to whether or not an initial condition thereof with respect to each of the operations when the program is executed can be satisfied by employing a predetermined analysis rule.
15 As a result, the log analyzing apparatus analyzes a cause of an error which happens to occur during the program execution, and then, provides this analysis result to the user.

Also, the log analyzing apparatus compares
20 the above-explained analysis result of the log information with an improvement rule indicative of a program improving method while the error happens to occur, and thus, provides such an improvement rule capable of improving the error, which is provided based
25 upon the analysis result of the log information.

As previously described, in accordance with the present invention, since the log information is analyzed by employing the predetermined analysis rule,

even when the user does not know the internal structure of the program which should be analyzed, the log analysis can be carried out. Also, the log analyzing apparatus can clarify the cause of the error based upon
5 the analysis result, and further, can improve this program.

As previously explained in detail, in accordance with the log analyzing apparatus of the present invention, the cause of the error occurred
10 while the program is executed is analyzed by judging as to whether or not each of the operations when the program is executed which are extracted from the log information can satisfy the initial condition thereof. As a consequence, even when the user does not know the
15 structure of the program, the log analysis can be carried out.

BRIEF DESCRIPTION OF THE DRAWINGS

A more better understanding of the present
20 invention may be made by reading a detailed description in conjunction with the accompanying drawings, in which:

Fig. 1 is a diagram for schematically representing a system arrangement of a log analyzing
25 apparatus according to an embodiment mode of the present invention;

Fig. 2 is a diagram for indicating one example of log information 103 which constitutes input

data of the log analyzing apparatus according to this embodiment mode;

Fig. 3 is a flow chart for describing a process sequence of a log acquiring/reading process operation executed by the log analyzing apparatus of this embodiment mode;

Fig. 4 is a diagram for indicating one example of an AP operation sequence and a log level definition 111 of the log analyzing apparatus according to this embodiment mode;

Fig. 5 is a diagram for indicating one example of a log table employed in the log analyzing apparatus according to this embodiment mode;

Fig. 6 is a diagram for representing one example of a user designation executed in a user designation command accepting process operation of the log analyzing apparatus according to this embodiment mode;

Fig. 7 is a flow chart for describing a process sequence of a log analyzing process operation executed in the log analyzing apparatus of this embodiment mode;

Fig. 8 is a diagram for indicating one example of an extracted result of log information obtained by a log analyzing process unit 108 of the log analyzing apparatus according to this embodiment mode;

Fig. 9 is a diagram for representing one example of an analysis rule 112 of the log analyzing

apparatus according to this embodiment mode;

Fig. 10 is a diagram for indicating a provision example of an analysis result obtained by a log analysis result providing process operation of the log analyzing apparatus according to this embodiment mode;

Fig. 11 is a flow chart for describing a process sequence of a program improvement supporting process operation executed in the log analyzing apparatus of this embodiment mode;

Fig. 12 is a diagram for showing one example of an improvement rule executed in the log analyzing apparatus according to this embodiment mode;

Fig. 13 is a diagram for representing a provision example of an improving measure executed in the program improvement supporting process operation of the log analyzing apparatus according to this embodiment mode; and

Fig. 14 is a diagram for indicating a continued portion of a provision example in such a case that a solving measure 1 is selected in the provision of improvement supporting process operation of the log analyzing apparatus according to this embodiment mode.

DESCRIPTION OF THE EMBODIMENTS

A description will now be made of a log analyzing apparatus, according to an embodiment mode of the present invention, which analyzes a log outputted

when a program is executed by a computer.

Fig. 1 is a schematic diagram for indicating a system arrangement of the log analyzing apparatus according to this embodiment mode. As indicated in 5 Fig. 1, a computer 2 of this embodiment mode contains a log analyzing unit 105, a log acquiring/reading process unit 106, a user designation condition accepting process unit 107, a log analyzing process unit 108, a log analysis result provision processing 10 unit 109, and a program improvement supporting process unit 110.

The log analyzing unit 105 corresponds to a process unit for controlling an entire analyzing process operation for logs. The log acquiring/reading 15 process unit 106 corresponds to such a process unit for acquiring log information 103 to read the acquired log information 103 which is outputted when a program is executed. The user designation condition accepting process unit 107 corresponds to such a process unit for 20 accepting a user designation condition indicative of log information to be extracted.

The log analyzing process unit 108 corresponds to a process unit which extracts information from such a log table indicative of the 25 content of the log information 103 based upon the accepted user designation condition. Also, this process operation judges as to whether or not each of operations while a program is executed can satisfy an

initial condition thereof in order to analyze a cause of an error occurred when the program is executed.

The log analysis result provision processing unit 109 corresponds to such a process unit for
5 providing the above-described analysis result to a user. The program improvement supporting unit 110 corresponds to such a process unit which compares the analysis result of the log information 103 with an improving rule representative of an improving method
10 for a program when an error occurs, and provides an improving rule used to improve the error, which is provided based upon the analysis result of the log information 103.

It is so assumed that such a program is
15 recorded on a magnetic medium such as a CD-ROM and then this recording medium is stored into a magnetic disk drive and the like, and thereafter, this program is loaded onto a memory so as to be executed. This program may cause the computer 2 to function as the log
20 analyzing unit 105, the log acquiring/reading process unit 106, the user designation condition accepting process unit 107, the log analyzing process unit 108, the log analysis result provision processing unit 109, and also the program improvement supporting process
25 unit 110. It should be noted that the above-described recording medium used to record thereon the program may be realized by any recording media other than a CD-ROM.

In this embodiment mode, the application 101

outputs the detailed log information 103 by the log output processing unit 102. While this log information 103 is employed as input data, the log analyzing unit 105 for performing the log analyzing process is

5 constituted by the log analyzing/reading process unit 106, the user designation condition accepting process unit 107, the log analyzing process unit 108, the log analysis result provision processing unit 109, and also the program improvement supporting process unit 110.

10 Also, while the log analyzing unit 105 executes the log analyzing process, this log analyzing unit 105 refers to an application (will be abbreviated as an "AP" hereinafter) operation sequence, a log level definition 111, an analysis rule 112, and an improvement rule 113,

15 whereas the log analysis result provision processing unit 109 provides an analysis result to a user via the input/output means 104. It should also be noted that the computer 1 and the computer 2 may be constituted by the same computer machines.

20 Fig. 2 is a diagram for representing an example of the log information 103 which constitutes input data of the log analyzing apparatus according to this embodiment mode. In this example, while a first column up to a fifth column show initial setting

25 information, initial information such as a usable log level and a log output formation, which is determined by the log output processing unit 102, is outputted. A sixth column up to a 12-th column correspond to log

information related to a processing unit "A" of an application. A 14-th column up to an 18-th column correspond to log information related to a processing unit "C" of the application. Also, a 20-th column
5 corresponds to such log information when the application is completed.

Fig. 3 is a flow chart for indicating a process sequence of a log acquiring/reading process operation executed in the log analyzing apparatus
10 according to this embodiment mode. As indicated in Fig. 3, at a step 311, the log acquiring/reading process unit 106 accepts a positional designation input of log information, and acquires the log information 103 as input data. As the positional designation of
15 the log information 103, there is such a method for designating both a directory and a file name within the same computer machine. Also, as the positional designation of the log information 103, in the case that the log information 103 is located on a computer
20 machine connected to the Internet, a file may be designated based upon an instruction capable of identifying this file, for example, URL (UNIFORM RESOURCE LOCATORS) may be designated
(HTTP://WWW.HITACHI.CO.JP/SERVER/MACHINE01/LOGDATA/FILE
25 1).

At a step 312, both an application operation sequence (will be referred to as an "AP operation sequence" hereinafter) and the log level definition 111

are read in the memory. At a step 313, such a log table 501 shown in Fig. 5 is formed by establishing a correspondence relationship between the AP operation sequence and the log level definition 111, which have been read in the memory. Then, this log table 501 is outputted to either the memory or the buffer.

Fig. 4 is a diagram for showing one example of the AP operation sequence and of the log level definition 111 in the log analyzing apparatus according to this embodiment mode. This example represents such a case that the application software corresponds to document process software.

A log level correspondence example 401 shows a correspondence portion between the log level definition 411 and an operation sequence 412 of the document process application. In this example, for instance, an access process operation to a document database is defined as a "level 1." This level corresponds to, namely LOG LEVEL = 1 appeared in the log information 103.

An AP operation sequence correspondence example 402 indicates such a correspondence portion between the operation sequence 412 of the document process application and the operation command 413. For example, as to the access process operation to the document database, such a correspondence relationship indicates that a connection process operation corresponds to a command "CONNECT", i.e., a plurality

of actual commands correspond to classifications of these actual commands. While both the AP operation sequence and the log level definition 111 are employed which have been defined in this manner, the log table
5 501 is formed from the log information 103 in the process operation defined at the step 313.

Fig. 5 is a diagram for indicating one example of the log table employed in the log analyzing apparatus of this embodiment mode. When correspondence
10 relationships are established with respect to the contents of the log information 103 by using both the AP operation sequence and the log level definition 111, the log table of Fig. 5 may be made. For example, a first column up to a fifth column of the log
15 information 103 correspond to application initial setting information; a sixth column up to a seventh column correspond to initial setting information of a process operation "A"; an eighth column corresponds to an access process operation (connect) to the document
20 database; a ninth column corresponds to newly forming of document data; and the like. As explained above, since the log table 501 is formed from the log information 103 to be displayed, even such a user who does not know a structure of a program can readily
25 understand the content of this log information 103.

Fig. 6 is a diagram for showing one example of the user designation executed in the user designation condition accepting process operation of

the log analyzing apparatus according to this embodiment mode. A user designation example 601 shows such an example that log information extracted from the log table 501 is designated. For instance, Fig. 6
5 represents such an example case that the user designation example 601 is displayed on a screen, and a user marks log information to be extracted.

With respect to such log information to be extracted, the user may select to designate this log
10 information based upon the operation sequence of the document process application. In this example, the user instructs to extract the log information with respect to an access process operation to a document database; an update process operation of a document
15 database; newly forming of document data; an access process operation to document data; and also a deletion of document data; whereas the user instructs not to extract the log information with respect to an
exclusive process operation of data; and an exclusive
20 process operation of a database. As a result, such data which is not extracted is neglected in the log analyzing process operation.

Fig. 7 is a flow chart for indicating a process sequence of a log analyzing process operation
25 executed by the log analyzing apparatus according to this embodiment mode.

As indicated in Fig. 7, at a step 711, the user designation condition accepting process unit 107

accepts as an analysis condition designating input value, such a value which is entered by a user as a user designation condition indicative of the log information extracted in the above-described manner.

5 At a step 712, the log analyzing process unit 108 judges as to whether or not the above-described analysis condition designating input value is changed from the previously-inputted analysis condition input value. When the analysis condition designating input value is changed from the previous-inputted analysis condition input value, the log analyzing process operation is advanced to a step 713. At this step 713, the log analyzing process unit 108 extracts relevant data from the log table 501 under this designated condition and then, outputs the extracted data to either the memory or the buffer.

Fig. 8 is a diagram for indicating one example of extracted results of the log information by the log analyzing process unit 108 of the log analyzing apparatus according to this embodiment mode. An extraction result example 801 indicates data extracted by the user designation example 601 shown in Fig. 6. In this case, an access process operation (connect) to a document database, newly forming of document data and an access process to a document database are extracted result in the process operation "A". And an access process operation (acquire) to a document database, error are extracted result in the process operation

"C". It should also be noted that even when no designation is issued from the user in this example, the error information is extracted.

At the next step 714, the log analyzing process unit 108 analyzes the above-explained extracted data based upon a program analysis rule 112, and provides the analysis result by the log analysis result provision processing unit 109. Thereafter, the log analyzing process operation is returned to the previous step 711 at which the user designation condition accepting process unit 107 accepts a next designation.

Fig. 9 is a diagram for indicating one example of the program analysis rule 112 employed in the log analyzing apparatus according to this embodiment mode. As illustrated in Fig. 9, the program analysis rule 112 represents an initial condition as to each of operations when the program is executed. For instance, as to a rule 1, in the case that an AP operation corresponds to an access process operation to document data, an initial condition of this access process operation determines that the log analyzing apparatus is previously connected to the document database.

When the log analyzing process unit 108 detects an error indicating an "abnormal end" by referring to the extracted result of the log information of a step 714, this log analyzing process unit 108 reads the initial condition of the previous

operation from the program analysis rule 112, and judges as to whether or not this read initial condition can be satisfied so as to execute the log analysis. For example, when the log analyzing process unit 108
5 detects "error: abnormal end by unauthorized process operation" in the extracted result example 801 of Fig. 8, this log analyzing process unit 108 reads out from the program analysis rule 112, such an initial operation "being previously connected to document
10 database" of the previous operation thereof "access process operation to document data", and thereafter, checks as to whether or not the log analyzing apparatus is connected to the document database. In other words, the log analyzing process unit 108 retrieves "access
15 process operation to document database" from the extracted result example 801 of Fig. 8, and confirms as to whether or not the log analyzing apparatus is connected to the document database. In this case, since "access process operation (disconnect) to
20 document database" is present subsequent to "access process operation (connect) to document database" within the extracted result example 801 of Fig. 8, and thereafter, "access process operation (acquire) to document data" is carried out, the log analyzing
25 process unit 108 judges that cutting of the document data may cause the abnormal end.

Fig. 10 is a diagram for showing a provision example of the analysis results obtained by the log

analysis result provision processing operation of the log analyzing apparatus according to this embodiment mode. In a provision example 1001 of Fig. 10, execution results of an application are provided. This provision example 1001 of Fig. 10 represents that since an event "abnormal end" happens to occur in the process operation "C", a document process application is ended under abnormal status.

In another provision example 1002, an analysis result obtained based upon the analysis rule 112 is provided as to the reason why the event "abnormal end" happens to occur. While the document data acquisition executed in the process operation "C" corresponds to the access operation to the document data, this provision example 1002 indicates that the connection to the document database corresponding to the initial condition used to execute this process operation is cut in the process operation "A" preceding to this process operation, and thereafter, no connection is made to the document database.

Fig. 11 is a flow chart for describing a process operation of a program improvement supporting process operation executed in the log analyzing apparatus according to this embodiment mode. At a step 1111 of Fig. 11, the program improvement supporting process unit 110 judges as to whether or not there is such an analysis result which is contradictory to the analysis rule 112 in the log analyzing process unit

108. When there is no analysis result which is contradictory to the analysis rule 112, this process operation is ended. To the contrary, when there is such an analysis result which is contradictory to the analysis rule 112, the program improvement supporting process unit 110 forms an applicable improvement measure with reference to an improvement rule 113 at a step 1112, and then provides the applicable improvement measure to the user.

10 At a step 1113, the program improvement supporting unit 110 confirms as to whether or not this improving measure can be agreed with the user. When this improving measure cannot be agreed with the user, the process operation is returned to the previous step 15 1112 at which the program improvement supporting process unit 110 newly forms an improving measure. At the step 1113 to the contrary, when the improving measure can be agreed with the user, the program improvement supporting process unit 110 outputs an 20 improvement history and records the improvement history at a step 1114, and thereafter, issues an instruction of improving the program.

 At a steps 1115, the program improvement supporting process unit 110 checks as to whether or not 25 such an improvement result obtained in the case that the relevant program is corrected in accordance with the above-explained provided improving measure is newly contradictory to the improvement rule 113. If this

improvement result is contradictory to the improvement rule 113, for instance, a shortage, or excess of contents of relative operation, then the process operation is returned to the previous step 1112. At this step 1112, the program improvement supporting process unit 110 provides an improvement measure. To the contrary, when the improving result is not contradictory to the improvement rule 113, the process operation is ended.

Fig. 12 is a diagram for showing one example of the improvement rule employed in the log analyzing apparatus according to the embodiment mode. For instance, in such a case that an error is detected from a portion of "operation" of an analysis rule, either "initial condition is inserted before operation" of the improvement rule 1 or "operation opposite to initial condition is deleted before operation" of the improvement rule 2 is provided as an improving measure.

Fig. 13 is a diagram for representing a provision example of the improving measure obtained by the program improvement supporting process operation of the log analyzing apparatus according to this embodiment mode. Both a provision example 1001 and another provision example 1002 of Fig. 13 correspond to the previously explained analysis results.

In a cause analysis of the provision example 1002, it is cleared that the connection to the document database in the process operation "A" is cut before the

data acquisition operation of the process operation
"C." As a result, in an improving measure 1 (namely,
provision example 1301) with employment of the
improvement rule 1, the user is inquired as to whether
5 or not the connection process operation to the document
database is inserted before the data acquisition
operation of the process operation "C". Also, in
another improving measure 2 (namely, provision example
1302) with employment of the improvement rule 2, the
10 user is inquired as to whether or not the connect/cut
process operation with respect to the document database
of the process operation "A" is deleted.

Fig. 14 is a diagram for indicating a
continuation of the provision example in the case that
15 the solving measure 1 is selected in the provision of
the improving measure obtained by the program
improvement supporting process operation of the log
analyzing apparatus according to this embodiment mode.
Since "YES" is selected in the improving measure 1 of
20 the provision example 1301, when the connection process
operation to the document database is inserted, a
shortage/excess of relative commends of an improvement
rule "N" occurs. A warning notice is made as follows:
A process operation is added also to the database cut
25 which is related to the database connect (provision
example 1401). It should be noted that this embodiment
mode has described only "database connect and database
cut" as a related rule "N", another warning notices may

be issued in such a case that a shortage/excess of other relative commands will occur.

As previously described in detail, in accordance with the log analyzing apparatus of this embodiment mode, the cause of the error occurred while the program is executed is analyzed by judging as to whether or not each of the operations when the program is executed which are extracted from the log information can satisfy the initial condition. As a result, even when the user does not know the structure of the program, the log analysis can be carried out.

Also, in accordance with the present invention, the cause of the error occurred while the program is executed is analyzed by judging as to whether or not each of the operations when the program is executed which are extracted from the log information can satisfy the initial condition. As a consequence, even when the user does not know the structure of the program, the log analysis can be carried out.